

那些天，我一直在思考一个问题：如何才能写出更好的代码？我尝试了各种方法，但似乎总是在同一个地方卡住。直到有一天，我遇到了一个让我豁然开朗的瞬间。

那是一个阳光明媚的下午，我正在办公室里敲代码。突然，我听到窗外传来一阵清脆的鸟鸣声。我停下手里的工作，走到窗边，看着窗外那片翠绿的草地和蓝天白云。那一刻，我仿佛找到了答案：好的代码应该是简洁、易读且易于维护的。它应该像这首诗一样，能够让人一眼就能看懂，并且能够随着时间和经验的增长而变得更加优秀。

从那以后，我开始更加注重代码的结构和逻辑。我会定期回顾自己的代码，找出可以优化的地方，并进行相应的修改。同时，我也开始学习更多的编程语言和技术，不断提升自己的技术水平。现在，我已经能够写出一些相对复杂的程序了，虽然还有很长的路要走，但我相信只要坚持下去，总有一天能够成为一名优秀的程序员。

```

        #include <iostream.h>
        class point {
            int x, y;
        public:
            point(int x=0, int y=0) { this->x=x; this->y=y; }
            void move(int dx, int dy) { x+=dx; y+=dy; }
            void print() const { cout << x << " " << y << endl; }
        };
        point p1(1,1), *pr;
        void movedx(int a, int b) { pr->move(a,b); }
        void movey(int a, int b) { pr->move(0,b); }
        void movez(int a, int b) { pr->move(a,0); }
        void print() const { cout << "x=" << p1.x << " y=" << p1.y << endl; }

        int main() {
            cout << "Input x,y: ";
            cin >> p1.x >> p1.y;
            cout << "Move dx,dy: ";
            cin >> movedx >> movey >> movez;
            cout << "Final position: ";
            p1.print();
            return 0;
        }
    
```


! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [] ^ _ ~ €
291, 2, 鸟 a c 拼音 鸟 a _ (u L L L 303, 16, 鸟 a 繁文 鸟 L L L 324, 28, 8 { | 婦 L L L 337, 40, P [鸟 a
347, 50, { | 婦 1 鸟 c 钯 钯 6 _ L C J O L C J _ 钯 溪 g 0 刻 弘 R ↑ _ □ | ! "# % & ' () * +, - ./ 0123456789 : ; < = ?
@ABCDEFHJKLNMOPQRSTUVWXYZ[] _ abcdefghijklmnopqrstuvwxyz { | }
~ €

胀腿.←搬□+, D|胀腿.←搬□+,
在屏幕上显示▲↑ZheJiangUniversityCS4061L 5LLL ;+Times New Roman|宋体 Arial
Wingdings|黑体■Soaring!第六章 类和对象16.10 对象指针和对象引用 幻灯片 3 幻灯片 4 幻灯片 5 幻灯片 6 幻灯片 7 幻灯片 8&6.10.2对象指针和对象引用作函数的参数 幻灯片 10 对象引用作函数参数 幻灯片 12 16.10.3 this 指针例Exa6-15 6.11 对象数组 幻灯片 16 6.11 指向数组的指针幻灯片 18 幻灯片 19 幻灯片 20 幻灯片 21 对象指针数组幻灯片 23 6.11.3带参数的main()函数幻灯片 25 幻灯片 26 6.12 常类型 一般常量 ■常对象 ■常指针 幻灯片 31 常引用幻灯片 33 常成员函数 幻灯片 35 常数据成员 幻灯片 37 6.13 子对象幻灯片 39 幻灯片 40 6.13.2 对堆区◀运算符新的用法 幻灯片 43 ▶运算符delete的用法 注意 幻灯片 46 幻灯片 47 6.14 类型转换 →构造函数具有类型转换功能 幻灯片 50 转换函数 幻灯片 52 |注意 + ▲已用的字体|▲演示文稿设计模板|▲ 幻灯片标题|5 81@# PID: HLINKS A